IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Brown et al.**	§	
	§	Group Art Unit: **2112**
Serial No.: **10/825,142**	§	
	§	Examiner: **Torres, Joseph D.**
Filed: **April 15, 2004**	§	
	§	Confirmation No.: **7822**
For: **Method and Apparatus for Supporting Checksum Offload in Partitioned Data Processing Systems**	§	

35525

PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on January 19, 2009.

A fee of $540.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

This appeal has no related proceedings or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1-39

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims canceled: 39

Claims withdrawn from consideration but not canceled: None

Claims pending: 1-38

Claims allowed: None

Claims rejected: 1-38

Claims objected to: None

C. CLAIMS ON APPEAL

The claims on appeal are: 1-38

STATUS OF AMENDMENTS

No Amendment was filed after final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

With the widespread use of networks, data packets are transmitted between different data processing systems. Each time a data packet is received, the data packet is processed before passing the data in the packet up to an application for use. In processing a data packet, a checksum calculation is performed to verify that the data in the data packet was transmitted without errors. When these checksum calculations are performed by the host processor or processors in a data processing system, the calculations are a processor intensive task. Many network adapters provide a task offload feature called checksum offload. With this feature, the network adapter calculates the needed checksums to verify whether the data packet is a good data packet. Additionally, checksum offload also provides for generating and adding checksums to data packets that are to be transmitted onto a network. In this manner, the processor resources of the host processor or processors are freed up for other uses.

The features of the present claims provide an improved method, apparatus, and computer instructions for processing data packets in a logical partitioned data processing system, by providing a checksum offload mechanism for use by partitions in a logical partitioned data processing system, in order to reduce the amount of checksum verifications that are performed in transporting data. In essence, an interpartition virtual network is formed through the use of virtual network adapters, where such data is transferred between the different partitions using virtual network adapters. In other words, an interpartition virtual network contains partitions, in a logical partitioned data processing system, that communicate with each other using virtual network adapters. In such an environment, it is not necessary to verify that data in data packets sent from one partition to another partition when the data originates within one of the partitions and

In addition, when a partition sends a data packet to another partition using a virtual network adapter, such partition does not need to generate a checksum if the data packet will eventually be sent over a physical network adapter and if that physical network adapter supports checksum offload. Further, verification of a checksum is unnecessary if the data packet arrived over a physical network adapter that supports checksum offload and the checksum is verified to be good. If a data packet is to be sent to any partition within the interpartition virtual network that does not support checksum offload, a checksum is generated. With respect to the component

that generates the checksum in these illustrative examples, the sending partition does not know whether a data packet will be delivered to a partition that supports checksum offload, one that does not, or bridged on to a physical network, when a data packet is to be sent by the sending partition. As a result, this partition is unable to decide whether to generate the checksum. As a result, the sending partition never generates a checksum. Instead, if a data packet is to be delivered to a partition that does not support checksum offload, platform firmware, such as a hypervisor, generates and adds a checksum to the data packet before delivering the data packet. The hypervisor performs this function because this component has knowledge of which virtual adapters support checksum offload and which virtual adapters do not support this feature in the illustrative examples. A data packet originating from outside the interpartition virtual network has its checksum verified at the final destination if the data packet was routed or bridged onto the interpartition virtual network without having the checksum being verified to be good by the physical adapter receiving the data packet.

The mechanisms of the present claimed features takes into account these situations by providing information in the form of flags for use in sending and receiving data packets within an interpartition virtual network in order to facilitate a checksum offloading technique that advantageously improves resource utilization.

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a method for processing a data packet in an interpartition virtual network in a logical partitioned data processing system (Specification page 4, lines 1-4; page 16, lines 1-15). The logical partitioned data processing system comprises a plurality of logical partitions that each simultaneously run an operating system therein (Specification page 13, line 4 – page 15, line 16; Figure 2, element 200), where each one of the logical partitions comprises a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical partitioned data processing system to thereby form the interpartition virtual network (Specification page 15, lines 17-38). The method comprises identifying a state of a first flag and a state of a second flag in the data packet in response to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system

(Specification page 4, lines 4-10); and selectively verifying a checksum, by the first partition in the logical partitioned data processing system, for the data packet as indicated by the state of the first flag and the state of the second flag (Specification page 4, lines 10-12; page 20, line 9 – page 21, line 20; Figure 4), where the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet (Specification page 19, lines 16-24; Figure 5, elements 506 and 508).

B. CLAIM 15 - INDEPENDENT

The subject matter of claim 15 is directed to a logical partitioned data processing system for processing a data packet in an interpartition virtual network in a logical partitioned data processing system (Specification page 4, lines 1-4; page 16, lines 1-15). The logical partitioned data processing system comprises a plurality of logical partitions that are each operable for simultaneously running an operating system therein (Specification page 13, line 4 – page 15, line 16; Figure 2, element 200), where each one of the logical partitions comprises a data processor for running its respective operating system and a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical partitioned data processing system to thereby form the interpartition virtual network (Specification page 15, lines 17-38). The data processing system comprises identifying means, responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, for identifying a state of a first flag and a state of a second flag in the data packet (Specification page 4, lines 4-10); and selectively verifying means, in a first partition in the logical partitioned data processing system, for selectively verifying a checksum for the data packet as indicated by the state of the first flag and the state of the second flag (Specification page 4, lines 10-12; page 20, line 9 – page 21, line 20; Figure 4), where the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet (Specification page 19, lines 16-24; Figure 5, elements 506 and 508).

The structure corresponding to the identifying means and the selectively verifying means is described in the Specification at page 13, line 4 – page 16, line 15, as depicted at element 200 of Figure 2.

C. CLAIM 29 - INDEPENDENT

The subject matter of claim 29 is directed to a computer program product recorded on a computer readable, recordable-type medium and operable for processing a data packet in an interpartition virtual network-by a logical partitioned data processing system (Specification page 4, lines 1-4; page 16, lines 1-15). The logical partitioned data processing system comprises a plurality of logical partitions that are each operable for simultaneously running an operating system therein (Specification page 13, line 4 – page 15, line 16; Figure 2, element 200), where each one of the logical partitions comprises a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical partitioned data processing system to thereby form the interpartition virtual network (Specification page 15, lines 17-38). The computer program product comprises first instructions, responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, for identifying a state of a first flag and a state of a second flag in the data packet (Specification page 4, lines 4-10); and second instructions for selectively verifying a checksum, by the first partition in the logical partitioned data processing system, for the data packet as indicated by the state of the first flag and the state of the second flag (Specification page 4, lines 10-12; page 20, line 9 – page 21, line 20; Figure 4), where the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet (Specification page 19, lines 16-24; Figure 5, elements 506 and 508).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

A. GROUND OF REJECTION 1

The rejection of Claim 28 under 35 U.S.C. § 112, second paragraph, as being incomplete for omitting essential elements;

B. GROUND OF REJECTION 2

The rejection of Claims 1, 15, 29, and 39 under 35 U.S.C. § 103 as being unpatentable over Thompson et al. (U.S. Patent No. 5,430,842) in view of Hefferon et al. (5,659,756) and Bean et al. (U.S. Patent No. 4,843,541); and

C. GROUND OF REJECTION 3

The rejection of Claims 1-39 under 35 U.S.C. § 103 as being unpatentable over Maezawa et al. (U.S. Patent No. 6,145,024) in view of Kondo et al. (U.S. Patent No. 6,618,396) and Lansing et al. (U.S. Publication No. 2003/0058862) in further view of Thompson et al. (U.S. Patent No. 5,430,842) in view of Hefferon et al. (5,659,756) and Bean et al. (U.S. Patent No. 4,843,541).

ARGUMENT

A. GROUND OF REJECTION 1 (Claim 28)

Claim 28 stands rejected under 35 U.S.C. § 112, second paragraph, as being incomplete for omitting essential elements.

1. *Claim 28*

The Examiner rejects Claim 28 under 35 U.S.C. § 112, second paragraph in stating that such claim is incomplete for omitting essential structural cooperation of the elements, since according to the Examiner Claim 28 *does not recite any structural elements connecting the data structures in Claim 28 to the steps of claim 15*.

Appellants urge that the reason Claim 28 does not recite structural elements connecting the data structure of Claim 28 to the steps of Claim 15 is because Claim 15 does not recite *any steps*, so it is not possible for Claim 28 to recite structural elements connecting the data structures in claim 28 to any alleged **steps** of claim 15. Claim 15 is instead directed to a data processing system.

It is further noted that Claim 28 expressly recites "wherein the first flag and the second flag are located in a header in the data packet". As can be seen, the features of Claim 28 pertain to where the two checksum-based flags are located, and in particular that such checksum-based flags are located in a header of *the data packet that is defined by the features of Claim 1* (per Claim 1: "A method for processing a **data packet** in an interpartition virtual network in a logical partitioned data processing system"; "responsive to receiving the **data packet** at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, identifying a state of a first flag and a state of a second flag in the **data packet**"; and "selectively verifying a checksum, by the first partition in the logical partitioned data processing system, for the **data packet** as indicated by the state of the first flag and the state of the second flag, wherein the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the **data packet**"). Thus, Claim 28 defines characteristics of the features recited in Claim 15, and therefore properly provides structural data packet characteristics for the **data packet** feature expressly recited in Claim 15, and therefore further limits the data processing system of Claim 15.

Thus, it is urged that Claim 28 has been erroneously rejected under 35 U.S.C. § 112, second paragraph since it does not omit essential elements, as alleged by the Examiner in rejecting such claim.

B. GROUND OF REJECTION 2 (Claims 1, 15, 29 and 39)

Claims 1, 15, 29, and 39 stand rejected under 35 U.S.C. § 103 as being unpatentable over Thompson et al. (U.S. Patent No. 5,430,842), hereinafter "Thompson" in view of Hefferon et al. (5,659,756), hereinafter "Hefferon" and Bean et al. (U.S. Patent No. 4,843,541), hereinafter "Bean".

1. *Claims 1, 15 and 29*

With respect to Claim 1, such claim recites "*responsive to receiving* the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, *identifying a state of a first flag and a state of a second flag in the data packet*" (emphasis added by Appellants). As can be seen, these features with respect to Claim 1 are associated with an action that occurs when a data packet is *received*, including the identification of the state of two flags that are in this received data packet. In rejecting this aspect of Claim 1, the Examiner states that Thompson teaches such data packet reception processing, citing Thompson col. 3, lines 40-49 and col. 8, lines 13-27. Appellants show that there, Thompson states:

> "In the inbound direction, network adapter 12 decodes the packet header and programs the checksum control information directly into internal registers. The network adapter 12 calculates the checksum as it transfers the packet to memory 11. When the network adapter 12 completes the calculation of the checksum, network adapter 12 appends the result to the data stream that is being transferred to the memory 11. The processor 15 compares this checksum result against the packet checksum to verify the data" (*Thompson col. 3, lines 40-49*) (emphasis added by Appellants); and

> "For example, FIG. 8 shows an *outbound packet 60 built in memory 11* (shown in FIG. 1). **Outbound packet** 60 includes a checksum control header 61, a link level header 62, an IP header 63, a transport header 64 and user data 65. Checksum control header is shown to include a start offset field 71, a stop offset field 75, an algo field 72, a direction field 73, an insert field 74 and an insert offset field 76. Start offset field 71 indicates the byte at which checksumming is to start. Stop

> offset field 75 indicates the stop offset, that is, the number of bytes which are to be checksummed. Algo field 72 indicates the checksum algorithm used (TCP, UDP, etc.). Direction field 73 indicates the direction of data flow (inbound or outbound). Insert field 74 indicates whether the ***outbound*** packet is to have a checksum ***inserted***. Insert offset field 76 indicates the location where a checksum is to be **inserted**" (*Thompson col. 8, lines 13-27*) (emphasis added by Appellants).

As to the cited passage at Thompson col. 3, such passage describes various operational steps that are performed when a data packet is *received*. The cited passage at Thompson col. 8, however, describes various operational steps that are performed when a data packet is to be *transmitted*. The receipt and transmission of data packets are separate and distinct operations, and steps describing operational steps with respect to *transmission* of data packets, such as those described by Thompson at col. 8, do not provide any description or teaching with respect to operational steps that occur upon *receiving* a data packet, which is what Claim 1 is directed to. Thus, Thompson's description at col. 8 does not provide any teaching as to any operational steps that occur "responsive to receiving" a data packet, as is recited in Claim 1. Thus, the Examiner's reliance on Thompson's teachings at col. 8 – which is *directed to steps that occur in anticipation of a data packet being transmitted* - does not provide any teaching/description of any operational steps that occur *responsive to receiving* the data packet, as per the features of Claim 1.

As to the cited Thompson passage at col. 3, which *is* directed to steps that occur when *receiving* a data packet, this description describes a traditional checksum calculation being unconditionally performed by the network adapter, including steps of decoding, programming, calculating, transferring, appending and comparing. Such traditional checksum calculation has already been expressly acknowledged by Appellants in the background section of their own Specification (page 1, lines 12-27; "Description of Related Art"). This unconditional checksum processing of received packets by a network adapter has been improved upon by the present invention, where the checksum is advantageously *selectively verified* – and such selective verification of the checksum occurs as indicated by the state of the first flag and the second flag. Thompson's teachings at col. 3 with respect to checksum processing *occurs unconditionally*, and such checksum processing is not described as being performed 'as indicated by the state of the first flag and the state of the second flag', as per the features of Claim 1. Instead, this cited passage states that the packet header is 'decoded', a checksum is 'calculated', which is then

'compared' with the received checksum in the received data packet. There is *no selective verification of the checksum of a received data packet as indicated by the state of* ***two flags that are received***, as per the features expressly recited in Claim 1.

Further with respect to Thompson's description at col. 8 regarding outbound packet data processing, to the extent such passage describes flags, these flags do not indicate whether to *selectively verify a checksum in a received data packet*, but instead *indicate whether a checksum is to be inserted in an outbound packet*. This is different from Claim 1 in that 'insertion' of a checksum is substantially different from 'verifying' a checksum, and processing associated with an 'outbound' packet has nothing to do with processing of an inbound/received data packet, as per the features of Claim 1. Nor are these flags described as being part of a data packet that is received, as per the features of Claim 1. Thus, it is further shown that Claim 1 has been erroneously rejected as Thompson's flags are not used to indicate selective *verification* of a *received* data packet, but instead indicates whether a checksum is to be *inserted* in an *outbound* data packet.

While it may be true that this checksum that is inserted by Thompson per the teachings at col. 8 may be subsequently used/checked in a checksumming operation, the state of the Thompson flags are not identified in response to receiving a data packet at a first partition, as claimed, and the state of such flags *are not used in determining whether to* ***selectively verify*** *a checksum in a received data packet*, as claimed, but instead are used to *indicate whether a checksum is to be inserted in an outbound packet*.

Nor do the other cited references to Hefferon and Bean overcome such teaching deficiency. For example, Hefferon does not describe any type of checksum processing. As further example, Bean does not describe any type of checksum processing.

In summary, Thompson's description of how it processes incoming packets does not teach any identification of checksum-based flags that are used for selectively verifying a checksum for the data packet as indicated by the state of the first flag and the state of the second flag. While Thompson describes use of flags in processing outgoing packets, these flags are used to *indicate whether a checksum is to be inserted in an outbound packet*. In contrast, Claim 1 recites use of checksum-based flags *in determining whether to selectively verify a checksum in a received data packet*. Any subsequent use of this checksum information, that Thompson inserts

in outgoing packets, is not described by any of the cited references as (1) identifying a state of two checksum-based flags in the data packet in response to receiving a data packet; or (2) selectively verifying a checksum, for the data packet as indicated by the state of the first flag and the state of the second flag, as claimed. Instead, such subsequent use – which is not described in the cited references – is reasonably assumed to be the process described by Thompson himself in the description of how incoming packets are processed by Thompson, and such incoming packet processing by Thompson does not describe the claimed use of the two checksum-based flags (identifying; selectively verifying), as previously shown in the Thompson col. 3 discussion hereinabove.

Thus, it is urged that Claim 1 has been erroneously rejected as the Examiner has failed to properly establish a prima facie showing of obviousness. [1]

Appellants further urge that a person of ordinary skill in the art would not have been motivated to modify the teachings of Thompson to include a *selective verification of a checksum of a received data packet*, as indicated by the state of two flags that identified in response to a data packet being received, because all of Thompson's packets are received across a traditional network, and accordingly all received packets have a checksum calculated by the network adapter as the adapter transfers the packet to memory (Thompson col. 3, lines 40-49; Figure 1, element 30). Thus, modifying Thompson in accordance with the above identified missing claimed features would merely add additional system complexity and processor overhead, with no associated benefit (per Claim 1, the benefit results in the use of these missing claimed features in a multi-partitioned data processing system where data packets internally flow between partitions of such multi-partitioned data processing system, which Thompson does not contemplate), as all Thompson packets need to be checked due to their arrival across a general network (Specification

[1] In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a prima facie case of obviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). Only if that burden is met, does the burden of coming forward with evidence or argument shift to the applicant. *Id.* All words in a claim must be considered in judging the patentability of that claim against the prior art." MPEP 2143.03; *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). **If the examiner fails to establish a prima facie case, the rejection is improper and will be overturned.** *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). In the absence of a proper *prima facie* case of obviousness, an applicant who complies with the other statutory requirements is entitled to a patent. *See In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992).

page 1, 1st paragraph in Section 2) - thus evidencing that a person of ordinary skill in the art would not have been motivated to modify Thompson to provide (1) responsive to receiving the data packet, identifying a state of a first flag and a state of a second flag in the data packet, where the first and second flags are checksum-based flags; or (2) selectively verifying a checksum, for the data packet as indicated by the state of the first flag and the state of the second flag, as claimed. Thus, it is further urged that Claim 1 has been erroneously rejected.

2. *Claim 39*

With respect to Claim 39, such claim has previously been cancelled (responsive to a Restriction Requirement dated September 17, 2007), and so it is unclear why Claim 39 is being rejected in the current Office Action dated October 21, 2008. It is urged that it is clear error to finally reject a previously cancelled claim, as such rejection clouds the file wrapper history of this case – and associated file wrapper estoppel and double-patenting issues - by casting doubt on whether such claim has been rejected, or alternatively cancelled based on a previous Restriction Requirement that stated such claim was directed to a separate invention.

C. GROUND OF REJECTION 3 (Claims 1-39)

Claims 1-39 stand rejected under 35 U.S.C. § 103 as being unpatentable over Maezawa et al. (U.S. Patent No. 6,145,024), hereinafter "Maezawa" in view of Kondo et al. (U.S. Patent No. 6,618,396), hereinafter "Kondo" and Lansing et al. (U.S. Publication No. 2003/0058862), hereinafter "Lansing" in further view of Thompson in view of Herron and Bean.

1. *Claims 1, 15 and 29*

Generally speaking with respect to Claim 1, it is urged that the combined teachings of the cited references describe a *conditional CRC check* based on a single ECC flag (Kondo), and a *conditional CRC generation* (but not a check) based on a single CRC flag (Lansing). Thus, the combination does not teach a *conditional CRC check* based on ***two*** different flags (a first flag *and* a second flag). It is further urged that none of the cited references teach or otherwise suggest sending data packets from one logical partition of a logical partitioned data processing system to another partition *within the same* logical partitioned data processing system *using virtual network*

adapters, as per the interpartition virtual network features provided by independent Claims 1, 15 and 29.

With respect to Claim 1, such claim recites "selectively verifying a checksum, by the first partition in the logical partitioned data processing system, for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet". As can be seen, such claimed features are directed to a *conditional CRC check* based on *two different flags*.

In rejecting one such CRC flag, the Examiner cites Lansing's teaching at step 9-5 in Figure 9 and Claim 1. Appellants urge that the Lansing flag is *not used in determining whether to verify a checksum of a received data packet*. Instead, the Lansing flag is used to determine *whether to generate a checksum* (Lansing Figure 9, element 905, the "Generate CRC Flag") *in an outgoing data packet*. Thus, even when the teachings of Kondo and Lansing are combined, such combination does not describe the *use of two flags in a selective checksum **verification** step*, as per the features of Claim 1. Instead, such resulting combination describes one flag (the Kondo flag) being used to indicate whether an error exists, and if so a CRC is verified, and the other flag (the Lansing flag) is used to determine whether to 'generate' a CRC (in contradistinction to 'verifying' a CRC, as per Claim 1). Quite simply, generating a CRC for an outgoing data packet, as described by Lansing and its *use of a flag as a part of such CRC generation* for an outgoing data packet, is very different from use of a flag in checksum verification for a received data packet, as claimed.

Further with respect to Claim 1, the cited Maezawa reference describes, as expressly acknowledged by the Examiner, that ***each** packet has a CRC checksum* used for verifying received data (see page 9 of the present Office Action dated July 3, 2008, lines 6-9), and thus it is urged that a person of ordinary skill in the art would not have been motivated to modify such Maezawa teachings of verifying *each* packet in accordance with the claimed feature of '*selectively* verifying' due to this Maezawa desire to verify each packet. Thus, the only motivation for making such a change must be coming from Appellants' own disclosure and

claims, which is impermissible hindsight analysis.[2]

In rebuttal, the Examiner now states (see page 12 of the present Office Action dated October 21, 2008) that:

> "both Thompson, Kondo and Lansing provide motivation for providing the particular motivation for using flags in the Kondo and Lansing patents".

This circular logic (provide motivation for providing particular motivation for the cited references to *themselves* use their own flags) does not comply with the KSR requirements for establishing obviousness. In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). "Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR Int'l. Co. v. Teleflex*, Inc., 550 U.S. 398 (2007). "Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006))." The Examiner assertion that Thompson, Kondo and Lansing provide motivation for providing the particular motivation for using flags in the Kondo and Lansing patents does not provide any rationale underpinning of why a person of ordinary skill in the art would have been motivated the primary reference to Maezawa to include usage of such flags, and Appellants have provided logical underpinnings as to why a person of ordinary skill in the art would *not* have been

[2] It is error to reconstruct the patentee's claimed invention from the prior art by using the patentee's claims as a "blueprint". When prior art references require selective combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight obtained from the invention itself. *Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 227 USPQ 543 (Fed. Cir. 1985).

motivated to make such a modification, as previously described.

Claim 1 also recites interpartition packet features. Specifically, Claim 1 recites "responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, identifying a state of a first flag and a state of a second flag in the data packet". As can be seen, the flag states are identified 'responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system'. None of the cited references teach receiving such an interpartition data packet, and thus none of the cited references teach an action being performed (e.g., 'identifying') that occurs 'responsive to receiving' such (missing) interpartition data packet. In rejecting this aspect of Claim 1, the Examiner states that Maezawa teaches receiving such an interpartition data packet. For example, on page 26 of the present Office Action, at the bottom of the page, the Examiner states:

> "Maezawa teaches receiving a data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system (col. 3, lines 60-65, col. 12, lines 1-12 and Figure 1)"

Appellants urge that the Maezawa description does not describe the claimed 'interpartition virtual network' since such reference does not describe "wherein each one of the logical partitions comprises a virtual network **adapter** that is used to send data packets to other virtual network **adapters** of other logical partitions within the logical partitioned data processing system *to thereby form the interpartition virtual network*". As Claim 1 explicitly defines the claimed 'interpartition virtual network' to be a network where each one of the logical partitions comprises *a virtual network adapter that is used to send data packets to other virtual network adapters* of other logical partitions within the logical partitioned data processing system, and the cited Maezawa reference does not describe such virtual network adapters or their associated claimed features, Maezawa cannot describe or teach the claimed interpartition virtual network. Accordingly, as Maezawa does not describe/teach the claimed interpartition virtual network, it necessarily logically follows that Maezawa also cannot describe/teach data packet transfer in

such a (missing) interpartition virtual network. Thus, contrary to the Examiner's assertion in rejecting Claim 1, Maezawa does not teach the claimed interpartition virtual network, or the receiving of a data packet at a first partition of such (missing) interpartition virtual network. Thus, it is further urged that Claim 1 has been erroneously rejected due to this additional prima facie obviousness deficiency.

In this same rebuttal section of the Final Office Action, the Examiner asserts at the bottom of page 13 that:

> "Maezawa teaches receiving a data packet at a first partition of the interpartition virtual network from a second partition in the interpartition virtual network"

citing Maezawa col. 3, lines 60-65, col. 12, lines 1-12 and Figure 1. Appellants urge that Claim 1 also recites that the selective checksum processing of packets between partitions of such interpartition virtual network ("responsive to receiving the data packet *at a first partition in the interpartition virtual network from a second partition* in the interpartition virtual network ..."). Maezawa does not describe *checksum processing for data packets received between partitions* of an interpartition virtual network. Instead, Maezawa describes receiving packets from *external Input/Output devices* across an optical fiber link (col. 8, lines 9-30; col. 11, lines 3-10; Figure 2, elements 21 and 33). The Maezawa *checksum processing* of Figure 5 is done with respect to these packets *received from external I/O devices across an optical fiber link*. This can be seen by Maezawa's description at col. 16, lines 12-30, where it states:

> Referring to FIG. 5, there are shown a transfer data frame format A 100 on the large-capacity interface (optical fiber protocol frame, which can be of variable length) and a transfer data frame format B 101 on the medium-capacity interface (a conventional format). Conversion of each field per frame between these formats is indicated by arrows in FIG. 5. Each arrow direction indicates a direction of conversion from the frame format B 101 to the frame format A 100. In FIG. 5, there is illustrated an example of block conversion to be performed **when a frame received from the input/output device 4** through the switching device 7 is transferred to the large-capacity interface through the multiplexer port 9. In contrast, a frame in the format A 100 from the multiplexer channel is converted by the multiplexer port 9 in the direction opposite that shown by the arrows, and then it is transferred to the medium-capacity interface through internal matrix switching in the switching device 7.

Processing of *externally received frames*, as depicted by Maezawa's Figure 5, does not describe processing of packets received between partitions of a virtual interpartion network, as claimed.

Further yet with respect to Claim 1, the Examiner has mischaracterized the teachings of the cited Thompson reference in their attempt to establish prima facie obviousness. The Examiner cites Thompson's col. 3, lines 40-49 and col. 8, lines 13-27 as teaching the claimed feature of *responsive to receiving a data packet* at a first partition from a second partition, identifying a state of two flags. Appellants urge clear error in such assertion, as will now be shown in detail.

As to the cited Thompson passage at col. 3, which is directed to steps that occur when receiving a data packet, this description describes a traditional checksum calculation being unconditionally performed by the network adapter, including steps of decoding, programming, calculating, transferring, appending and comparing. Such traditional checksum calculation has already been expressly acknowledged by Appellants in the background section of their own Specification (page 1, lines 12-27; "Description of Related Art"). The operational steps with respect to *transmission* of data packets, such as those described by Thompson at the cited col. 8 passage, do not provide any description or teaching with respect to operational steps that occur upon *receiving* a data packet, which is what Claim 1 is directed to. Quite simply, Thompson's description at col. 8 does not provide any teaching as to any checksum-based flag identification or selective checksum verification operational steps that occur "responsive to receiving" a data packet, as is recited in Claim 1. The Examiner's reliance on Thompson's teachings at col. 8 – which is directed to steps that occur in anticipation of a data packet being *transmitted* - does not provide any teaching/description of any (i) checksum-based flag identification or (ii) selective checksum verification operational steps that occur *responsive to receiving* the data packet, as per the features of Claim 1. Thus, the Examiner's characterization of the Thompson teachings is clearly erroneous – further evidencing that Claim 1 has been erroneously rejected.

2. *Claims 2, 16 and 30*

Appellants initially urge error in the rejection of Claim 2 for reasons given above with respect to Claim 1 (of which Claim 2 depends upon).

Further with respect to Claim 2, it is urged that none of the cited references teach or suggest the claimed feature of "wherein the first flag is a no checksum flag that is used by the selectively verifying step to determine whether or not there is a checksum value included in the data packet that is received and *the second flag is a checksum good flag that is used by the selectively verifying step to determine whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received*". As can be seen, the features of Claim 2 are directed to features/characteristics associated with the two flags. Specifically, the first flag is a no checksum flag that is used by the selectively verifying step to determine whether or not there is a checksum value included in the data packet that is received. The second flag is a checksum good flag that is used by the selectively verifying step to determine whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received. In rejecting Claim 2, the Examiner cites Kondo col. 39, lines 55-67 and Lansing's step 905 of Figure 9 as teachings the features of Claim 2. Appellants urge clear error in such assertion, as will now be described in detail.

Kondo describes at col. 39, line 55-67 an ECC flag that indicates whether or not there is an ECC error. Such ECC flag: **(1)** is not used by a selectively verifying step to *determine whether or not there is a checksum value included in the data packet that is received* (and thus this ECC flag is not equivalent to the claimed first flag) – instead this flag indicates if there is an *error* (or not), and **(2)** is not used by a selectively verifying step to determine whether or not the data packet *has previously been verified as being good* based on a checksum included in the data packet that is received (and thus this ECC flag is not equivalent to the claimed second flag) - instead this ECC flag indicates if there is an *error* (or not).

Lansing describes at Figure 9, element 905 a CRC generation flag that indicates *whether or not a CRC is to be generated during packet transmission*. Such CRC generation flag (1) is not used by a selectively verifying step to *determine whether or not there is a checksum value included in the data packet that is received* (and thus this CRC generation flag is not equivalent to the claimed first flag) – instead this flag indicates if a *CRC should be generated* for a packet to be *transmitted*, and (2) is not used by a selectively verifying step to determine whether or not the data packet *has previously been verified as being good* based on a checksum included in the data packet that is received (and thus this CRC generation flag is not equivalent to the claimed second

flag) - instead this CRC generation flag indicates if *CRC should be generated* for a packet to be *transmitted*.

Thus, neither of these cited passages describes *either one* of the two flags expressly recited in Claim 2, and therefore it is further urged that Claim 2 (and dependent Claims 3-10) has been erroneously rejected due to this additional prima facie obviousness deficiency.

3. *Claims 3, 17 and 31*

Appellants initially urge error in the rejection of Claim 3 (and similarly for Claims 17 and 31) for reasons given above with respect to Claim 2 (of which Claim 3 depends upon).

Further with respect to Claim 3, such claim recites "wherein the selectively verifying step includes: verifying the checksum for the data packet only if the first flag and the second flag are unset". As can be seen, the features of Claim 3 are directed to a further refinement of the selectively verifying step of Claim 1, in that the checksum is only verified if both the first flag and the second flag are unset.

In rejecting Claim 3, the Examiner cites the Kondo flag and the Lansing flag as teaching two flags. As previously shown, the Lansing flag indicates *whether or not a CRC is to be generated during packet transmission* – and does not control any type of verification processing. Thus, the reliance on the Lansing flag as being one of the two flags recited in Claim 3 is clearly erroneous, as such flag is not used to control checksum verification, and therefore the resulting combination does not teach/suggest "wherein the selectively verifying step includes: ***verifying the checksum*** for the data packet *only if the first flag and the second flag are unset*" (emphasis added by Appellants), as expressly recited in Claim 3. Therefore, it is further urged that Claim 3 has been erroneously rejected due to this additional prima facie obviousness deficiency.

4. *Claims 4, 18 and 32*

Appellants initially urge error in the rejection of Claim 4 (and similarly for Claims 18 and 32) for reasons given above with respect to Claim 2 (of which Claim 4 depends upon).

Further with respect to Claim 4 (and similarly for Claims 18 and 32), such claim recites "wherein the selectively verifying step includes: skipping verification of the checksum if the first flag is set". As can be seen, the features of Claim 4 are directed to particulars associated with the

selectively verifying step, where *checksum verification is skipped if the first flag is set*. The Examiner asserts that Lansing teaches such selectively verifying details at Figure 9, elements 905-915. Appellants urge clear error, as this cited Lansing passage: **(1)** is not directed to any type of *verification* step, but instead is directed to a *generation* step (Lansing Figure 9, element 910), and **(2)** does not describe any skipping of checksum *verification*, but instead is directed to skipping a checksum *generation* step (Lansing Figure 9, element 910). Thus, contrary to the Examiner's assertion, Lansing does not teach the features expressly recited in Claim 4, and therefore it is further urged that Claim 4 has been erroneously rejected due to this additional prima facie obviousness deficiency.

It is further noted that Claim 4 is not directed to conditionally performing checksumming based upon whether or not a packet has a checksum included in it (which, in any event, is not taught by any of the cited references), but instead is directed to selective checksum verification based in the state of the checksum-based flag – which is a separately claimed element from the claimed checksum.

5. *Claim 6*

Appellants initially urge error in the rejection of Claim 6 for reasons given above with respect to Claim 2 (of which Claim 6 depends upon).

Further with respect to Claim 6, such claim recites "wherein the second flag is conditionally unset by the logical partitioned data processing system if the packet was received through a first virtual adapter associated with the first partition". As can be seen, the features of Claim 6 are specifically directed to a conditional unsetting of the second flag, where such flag unsetting condition is "if the packet was received through a first virtual adapter associated with the first partition".

In the new reason given in rejecting Claim 6, the Examiner now asserts:

> "Kondo and Lansing teach adaptive parameters (Steps 905-915 in Figure 9 of Lansing teaches a first CRC flag used to indicate the presence of redundancy; Col. 39, lines 62-67 in Kondo teaches a second ECC flag in a packet indicating whether errors are present) for allowing a sending station to notify a receiving station whether a transmitted packet has redundancy for use in verifying the packet based on a first CRC flag and whether the packet has errors so the

receiving controller can imitate error handling based on a second ECC flag intended for use in the particular embodiments of claims 6-10, 14, 20-25, 28 and 34-38."

Appellants urge that such broad-brushed analysis with respect to Claims 6-10, 14, 20-25, 28 and 34-38 has failed to properly establish prima facie obviousness with respect to the particular claimed features expressly recited in Claim 6. For example, an assertion of 'for use in verifying the packet' and "whether the packet has errors" does not establish any teaching/suggestion with respect to *conditionally unsetting a flag* if the packet was received through a first virtual adapter associated with the first partition, as expressly recited in Claim 6. Therefore, it is further urged that Claim 6 has been erroneously rejected due to this additional prima facie obviousness deficiency.

6. *Claim 7*

Appellants initially urge error in the rejection of Claim 7 for reasons given above with respect to Claim 2 (of which Claim 7 depends upon).

Further with respect to Claim 7, such claim recites "wherein the first flag is conditionally set by the logical partitioned data processing system if the data packet, received from the second partition, originated from within the logical partitioned data processing system". As can be seen, the features of Claim 7 are specifically directed to a conditional setting of the first flag, where such condition is "if the data packet, received from the second partition, originated from within the logical partitioned data processing system". The Examiner makes no assertion as to any teaching/suggestion with respect to the condition regarding *where the data packet originated* that is used to conditionally set the first flag. Therefore, it is further urged that Claim 7 has been erroneously rejected due to this additional prima facie obviousness deficiency.

7. *Claim 8*

Appellants initially urge error in the rejection of Claim 8 for reasons given above with respect to Claim 2 (of which Claim 8 depends upon).

Further with respect to Claim 8, such claim recites "wherein the second flag is conditionally unset by the logical partitioned data processing system if the data packet, received

from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked". As can be seen, the features of Claim 8 are specifically directed to conditionally unsetting the second flag, where such second flag unsetting condition is "if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked". The Examiner makes no assertion as to any teaching with respect to the *conditional unsetting of the second flag* if the data packet, received from the second partition, *was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked*. Therefore, it is further urged that Claim 8 has been erroneously rejected due to this additional prima facie obviousness deficiency.

8. ***Claim 9***

Appellants initially urge error in the rejection of Claim 9 for reasons given above with respect to Claim 8 (of which Claim 9 depends upon).

Further with respect to Claim 9, such claim recites "wherein the first flag is conditionally unset by the logical partitioned data processing system and the second flag is conditionally unset by the logical partitioned data processing system if the data packet was received by a physical network adapter that (i) is associated with the second partition, and (ii) does not support checksum offload". As can be seen, the features of Claim 9 are specifically directed to a conditional unsetting of the first and second flags, where such first and second flag unsetting condition is "if the data packet was received by a physical network adapter that (i) is associated with the second partition, and (ii) does not support checksum offload". The Examiner makes no assertion as to any teaching with respect to the conditional unsetting of the first flag or the conditional unsetting of the second flag if the data packet was received by a physical network adapter that **(i)** is associated with the second partition, and **(ii)** does not support checksum offload". Therefore, it is further urged that Claim 9 has been erroneously rejected due to this additional prima facie obviousness deficiency.

9. *Claim 10*

Appellants initially urge error in the rejection of Claim 10 for reasons given above with respect to Claim 8 (of which Claim 10 depends upon).

Further with respect to Claim 10, such claim recites "wherein the first flag is conditionally unset by the logical partitioned data processing system and the second flag is conditionally set by the logical partitioned data processing system if a physical adapter, supporting a checksum offload, verified the checksum as being good". As can be seen, the features of Claim 10 are specifically directed to a conditional unsetting of the first flag and conditional setting of the second flag, where such first flag unsetting condition and second flag setting condition is "if a physical adapter, supporting a checksum offload, verified the checksum as being good". The Examiner makes no assertion as to any teaching with respect to a conditional unsetting of the first flag nor a conditional setting of the second flag if a physical adapter, supporting a checksum offload, *verified the checksum as being good*. Therefore, it is further urged that Claim 10 has been erroneously rejected due to this additional prima facie obviousness deficiency.

10. *Claim 11*

Appellants initially urge error in the rejection of Claim 11 for reasons given above with respect to Claim 1 (of which Claim 11 depends upon).

Further with respect to Claim 11, such claim recites "wherein the first virtual adapter is a software device driver that has no associated physical hardware". In rejecting Claim 11, the Examiner states that Maezawa teaches the features of Claim 11 by Maezawa's multiplexor channel devices 3 and 10 in Figure 1. Appellants urge that such physical hardware devices do not teach "wherein the first virtual adapter *is a software device driver that has no associated physical hardware*", as per the features of Claim 11. Therefore, it is further urged that Claim 11 has been erroneously rejected due to this additional prima facie obviousness deficiency.

In an apparent acknowledgement that this missing claimed feature is not taught by the cited references, the Examiner's new position now is that the claimed software device driver is merely an intended use. Appellants urge error in such assertion, and such claim does not recite any 'intent' or 'use'. Instead, such claim positively recites that the first virtual adapter *is a*

software device driver – which is not a mere intended use, but a thing (device driver).

11. ***Claims 12 and 26***

Appellants initially urge error in the rejection of Claim 12 (and similarly for Claim 26) for reasons given above with respect to Claim 1 (of which Claim 12 depends upon).

Further with respect to Claim 12 (and similarly for Claim 26), such claim recites "conditionally generating the checksum for the new data packet only if the new data packet is to be sent outside of the interpartition virtual network by a physical network adapter". As can be seen, the checksum generation is conditional, with such checksum generation condition being "*only if* the new data packet is to be sent *outside* of the interpartition virtual network by a physical network adapter". In rejecting this aspect of Claim 12, the Examiner cites Maezawa's circuit 37 in Figure 2 as teaching such claimed limitation. Appellants urge clear error, as this circuit 37 is not described as generating checksums, as per the features of Claim 12, but instead is used for transmitting frames (Maezawa col. 11, lines 27-32). Transmitting of frames, as described by Maezawa does not teach or suggest "conditionally generating the checksum for the new data packet *only if the new data packet is to be sent outside of the interpartition virtual network by a physical network adapter*", as claimed. Therefore, it is further urged that Claim 12 has been erroneously rejected due to this additional prima facie obviousness deficiency.

12. ***Claims 13 and 27***

Appellants initially urge error in the rejection of Claim 13 (and similarly for Claim 27) for reasons given above with respect to Claim 1 (of which Claim 13 depends upon).

Further with respect to Claim 13 (and similarly for Claim 27), such claim recites "wherein the first flag and the second flag are added to a header of the data packet that is received by firmware of the logical data processing system during routing of the data packet, to a given partition of the logical partitioned data processing system, by the firmware". As can be seen, the features of Claim 13 are specifically directed to a particular technique for adding flags to a header of a data packet. In rejecting Claim 13, the Examiner states on the bottom of page 36 of the present Office Action that Maezawa teaches a 'means for sending' by Maezawa's circuits 37 and 38 of Figure 2:

> "Maezawa teaches **means for sending** the new data packet to the target destination using one of the physical network adapter (external interface protocol control circuit 37 in Figure 2 of Maezawa) or a virtual network adapter (link connection control circuit 38 in Figure 2 of Maezawa) (bottom of page 36 of the present Office Action dated October 21, 2008).

Appellants urge clear error, as Claim 13 recites data packet header operations in combination with firmware, and *does not recite any type* ***of means for sending***, as alleged by the Examiner in rejecting Claim 13. Instead, Claim 13 recites:

> wherein the first flag and the second flag are **added to a header of the data packet that is received by firmware** of the logical data processing system **during routing** of the data packet, to a given partition of the logical partitioned data processing system, **by the firmware**

None of the cited references describe **(1)** firmware of a logical data processing system that (i) *receives* a data packet and (ii) routes the data packet, or **(2)** where *during such routing* of a *received* data packet by the firmware, checksum-based flags are added to a header of such data packet. For example, Thompson describes inserting checksum information into packets being *transmitted* (in contrast, the features of Claim 13 are directed to actions performed on a *received* data packet), and this Thompson checksum information is not two flags that are used in determining whether or not to selectively verify a checksum in a received data packet. Nor does Thompson describe adding flags to a header of a data packet during routing of a received packet by firmware. Therefore, it is further urged that Claim 13 has been erroneously rejected due to this additional prima facie obviousness deficiency.

13. ***Claims 20 and 34***

Appellants initially urge error in the rejection of Claim 20 (and similarly for Claim 34) for reasons given above with respect to Claim 16 (of which Claim 20 depends upon).

Further with respect to Claim 20 (and similarly for Claim 34), such claim recites "wherein the second flag is conditionally unset by the logical partitioned data processing system if the packet was received through a first virtual adapter associated with the first partition". As

can be seen, the features of Claim 20 are specifically directed to a conditional unsetting of the second flag, where such flag unsetting condition is "if the packet was received through a first virtual adapter associated with the first partition".

In the new reason given in rejecting Claim 20, the Examiner now asserts:

> "Kondo and Lansing teach adaptive parameters (Steps 905-915 in Figure 9 of Lansing teaches a first CRC flag used to indicate the presence of redundancy; Col. 39, lines 62-67 in Kondo teaches a second ECC flag in a packet indicating whether errors are present) for allowing a sending station to notify a receiving station whether a transmitted packet has redundancy for use in verifying the packet based on a first CRC flag and whether the packet has errors so the receiving controller can imitate error handling based on a second ECC flag intended for use in the particular embodiments of claims 6-10, 14, 20-25, 28 and 34-38."

Appellants urge that such broad-brushed analysis with respect to Claims 6-10, 14, 20-25, 28 and 34-38 has failed to properly establish prima facie obviousness with respect to the particular features expressly recited in Claim 20. For example, an assertion of 'for use in verifying the packet' and "whether the packet has errors" does not establish any teaching with respect to *conditionally unsetting a flag* if the packet was received through a first virtual adapter associated with the first partition, as expressly recited in Claim 20. Therefore, it is further urged that Claim 20 has been erroneously rejected due to this additional prima facie obviousness deficiency.

14. ***Claims 21 and 35***

Appellants initially urge error in the rejection of Claim 21 (and similarly for Claim 35) for reasons given above with respect to Claim 16 (of which Claim 21 depends upon).

Further with respect to Claim 21 (and similarly for Claim 35), such claim recites "wherein the first flag is conditionally set by the logical partitioned data processing system if the data packet, received from the second partition, originated from within the logical partitioned data processing system". As can be seen, the features of Claim 21 are specifically directed to a conditional setting of the first flag, where such first flag setting condition is "if the data packet, received from the second partition, originated from within the logical partitioned data processing system". The Examiner makes no assertion as to any teaching/suggestion with respect to the

condition regarding *where the data packet originated* that is used to conditionally set the first flag. Therefore, it is further urged that Claim 21 has been erroneously rejected due to this additional prima facie obviousness deficiency.

15. *Claims 22 and 36*

Appellants initially urge error in the rejection of Claim 22 (and similarly for Claim 36) for reasons given above with respect to Claim 16 (of which Claim 22 depends upon).

Further with respect to Claim 22 (and similarly for Claim 36), such claim recites "wherein the second flag is conditionally unset by the logical partitioned data processing system if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked". As can be seen, the features of Claim 22 are specifically directed to conditionally unsetting the second flag, where such second flag unsetting condition is "if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked". The Examiner makes no assertion as to any teaching with respect to the *conditional unsetting of the second flag* if the data packet, received from the second partition, *was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked*. Therefore, it is further urged that Claim 22 has been erroneously rejected due to this additional prima facie obviousness deficiency.

16. *Claims 23 and 37*

Appellants initially urge error in the rejection of Claim 23 (and similarly for Claim 37) for reasons given above with respect to Claim 22 (of which Claim 23 depends upon).

Further with respect to Claim 23 (and similarly for Claim 37), such claim recites "wherein the first flag is conditionally unset by the logical partitioned data processing system and the second flag is conditionally unset by the logical partitioned data processing system if the data packet was received by a physical network adapter that (i) is associated with the second partition, and (ii) does not support checksum offload". As can be seen, the features of Claim 23 are specifically directed to a conditional unsetting of the first and second flags, where such first and

second unsetting condition is "if the data packet was received by a physical network adapter that (i) is associated with the second partition, and (ii) does not support checksum offload". The Examiner makes no assertion as to any teaching with respect to the conditional unsetting of the first flag or the conditional unsetting of the second flag if the data packet was received by a physical network adapter that **(i)** is associated with the second partition, and **(ii)** does not support checksum offload". Therefore, it is further urged that Claim 23 has been erroneously rejected due to this additional prima facie obviousness deficiency.

17. ***Claims 24 and 38***

Appellants initially urge error in the rejection of Claim 24 (and similarly for Claim 38) for reasons given above with respect to Claim 22 (of which Claim 24 depends upon).

Further with respect to Claim 24 (and similarly for Claim 38), such claim recites "wherein the first flag is conditionally unset by the logical partitioned data processing system and the second flag is conditionally set by the logical partitioned data processing system if a physical adapter, supporting a checksum offload, verified the checksum as being good". As can be seen, the features of Claim 24 are specifically directed to a conditional unsetting of the first flag and conditional setting of the second flag, where such unsetting of the first flag and setting of the second flag condition is "if a physical adapter, supporting a checksum offload, verified the checksum as being good". The Examiner makes no assertion as to any teaching with respect to a conditional unsetting of the first flag if a physical adapter, supporting a checksum offload, *verified the checksum as being good*. Therefore, it is further urged that Claim 24 has been erroneously rejected due to this additional prima facie obviousness deficiency.

18. ***Claim 25***

Appellants initially urge error in the rejection of Claim 25 for reasons given above with respect to Claim 15 (of which Claim 25 depends upon).

Further with respect to Claim 25, such claim recites "wherein the first virtual adapter is a software device driver that has no associated physical hardware". None of the cited references teach or suggest "wherein the first virtual adapter *is a software device driver that has no associated physical hardware*", as per the features of Claim 25. Therefore, it is further urged that

Claim 25 has been erroneously rejected due to this additional prima facie obviousness deficiency.

This claim is being separately argued from Claim 11 due to the Examiner's erroneous intended-use exception to method claims, as Claim 25 is a system claim. Thus, even assuming arguendo that there is an intended-use exception for method claims – which Appellants deny, as such 'no patentable weight' in a *process/method* claim would fly in the face of the *In re Bilski* machine-or-transformation requirement for process claims [3] – such reasoning does not apply to system/apparatus claims such as is provided by Claim 25.

19. ***Claim 39***

With respect to Claim 39, such claim has previously been cancelled (responsive to a Restriction Requirement dated September 17, 2007), and so it is unclear why Claim 39 is being rejected in the current Office Action dated October 21, 2008. It is urged that it is clear error to finally reject a previously cancelled claim, as such rejection clouds the file wrapper history of this case – and associated file wrapper estoppel and double-patenting issues - by casting doubt on whether such claim has been rejected or, alternatively, cancelled based on a previous Restriction Requirement that stated such claim was directed to a separate invention.

---

[3] The Supreme Court … has enunciated a definitive test to determine whether **a process claim** is tailored narrowly enough to encompass only a particular application of a fundamental principle rather than to pre-empt the principle itself. A claimed process is surely patent-eligible under § 101 if: (1) **it is tied to a particular machine or apparatus**, or (2) it transforms a particular article into a different state or thing. *In re Bilski*, (No. 2007-1130, Fed. Cir. October 2008)(en banc).

D. CONCLUSION

As shown above, the Examiner has failed to state valid rejections against any of the claims. Therefore, Appellants request that the Board of Patent Appeals and Interferences reverse the rejections. Additionally, Appellants request that the Board direct the Examiner to allow the claims.

Date: March 19, 2009

Respectfully submitted,

/Wayne P. Bailey/

Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A method for processing a data packet in an interpartition virtual network in a logical partitioned data processing system, wherein the logical partitioned data processing system comprises a plurality of logical partitions that each simultaneously run an operating system therein, wherein each one of the logical partitions comprises a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical partitioned data processing system to thereby form the interpartition virtual network, the method comprising:

responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, identifying a state of a first flag and a state of a second flag in the data packet; and

selectively verifying a checksum, by the first partition in the logical partitioned data processing system, for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet.

2. The method of claim 1, wherein the first flag is a no checksum flag that is used by the selectively verifying step to determine whether or not there is a checksum value included in the data packet that is received and the second flag is a checksum good flag that is used by the

selectively verifying step to determine whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received.

3. The method of claim 2, wherein the selectively verifying step includes:

verifying the checksum for the data packet only if the first flag and the second flag are unset.

4. The method of claim 2, wherein the selectively verifying step includes:

skipping verification of the checksum if the first flag is set.

5. The method of claim 2, wherein the selectively verifying step includes:

skipping verification of the checksum for the data packet if the first flag is unset and the second flag is set.

6. The method of claim 2, wherein the second flag is conditionally unset by the logical partitioned data processing system if the packet was received through a first virtual adapter associated with the first partition.

7. The method of claim 2, wherein the first flag is conditionally set by the logical partitioned data processing system if the data packet, received from the second partition, originated from within the logical partitioned data processing system.

8. The method of claim 2, wherein the second flag is conditionally unset by the logical partitioned data processing system if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked.

9. The method of claim 8, wherein the first flag is conditionally unset by the logical partitioned data processing system and the second flag is conditionally unset by the logical partitioned data processing system if the data packet was received by a physical network adapter that (i) is associated with the second partition, and (ii) does not support checksum offload.

10. The method of claim 8, wherein the first flag is conditionally unset by the logical partitioned data processing system and the second flag is conditionally set by the logical partitioned data processing system if a physical adapter, supporting a checksum offload, verified the checksum as being good.

11. (Previously Presented) The method of claim 1, wherein the data packet is received by a first virtual adapter in the first partition connected to the interpartition virtual network, wherein the first virtual adapter is a software device driver that has no associated physical hardware.

12. The method of claim 1 further comprising:

generating a new data packet for a target destination;

conditionally generating the checksum for the new data packet only if the new data packet is to be sent outside of the interpartition virtual network by a physical network adapter; and

sending the new data packet to the target destination.

13. The method of claim 1, wherein the first flag and the second flag are added to a header of the data packet that is received by firmware of the logical data processing system during routing of the data packet, to a given partition of the logical partitioned data processing system, by the firmware.

14. The method of claim 1, wherein the identifying step comprises identifying a state of a first flag and a state of a second flag in a header in the data packet.

15. A logical partitioned data processing system for processing a data packet in an interpartition virtual network in a logical partitioned data processing system, wherein the logical partitioned data processing system comprises a plurality of logical partitions that are each operable for simultaneously running an operating system therein, wherein each one of the logical partitions comprises a data processor for running its respective operating system and a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical partitioned data processing system to thereby form the interpartition virtual network, the data processing system comprising:

identifying means, responsive to receiving the data packet at a first partition in the

interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, for identifying a state of a first flag and a state of a second flag in the data packet; and

selectively verifying means, in a first partition in the logical partitioned data processing system, for selectively verifying a checksum for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet.

16. The data processing system of claim 15, wherein the selectively verifying means comprises means for selectively verifying a checksum for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag is a no checksum flag located in the data packet that indicates to the selectively verifying means whether or not there is a checksum value included in the data packet that is received, and the second flag is a checksum good flag located in the data packet that indicates to the selectively verifying means whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received.

17. The data processing system of claim 16, wherein the selectively verifying means includes:

verifying means for verifying the checksum for the data packet only if the first flag and the second flag are unset.

18. The data processing system of claim 16, wherein the selectively verifying means includes:

skipping means for skipping verification of the checksum if the first flag is set.

19. The data processing system of claim 16, wherein the selectively verifying means includes:

skipping means for skipping verification of the checksum for the data packet if the first flag is unset and the second flag is set.

20. The data processing system of claim 16, further comprising means for conditionally unsetting the second flag if the packet was received through a first virtual adapter associated with the first partition.

21. The data processing system of claim 16, further comprising means for conditionally setting the first flag if the data packet, received from the second partition, originated from within the logical partitioned data processing system.

22. The data processing system of claim 16, further comprising means for conditionally unsetting the second flag if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked.

23. The data processing system of claim 22, further comprising means for conditionally unsetting the first flag and the second flag if the data packet was received by a physical network adapter associated with the second partition.

24. The data processing system of claim 22, further comprising means for conditionally unsetting the first flag and conditionally setting the second flag if a physical network adapter, supporting a checksum offload, verified the checksum as being good.

25. The data processing system of claim 15, further comprising virtual adapter means for receiving the data packet in the first partition connected to the interpartition virtual network, wherein the virtual adapter means is a software device driver that has no associated physical hardware.

26. The data processing system of claim 15 further comprising:

first generating means for generating a new data packet for a target destination;

second generating means for conditionally generating the checksum for the new data packet only if the new data packet is to be sent outside of the interpartition virtual network by a physical network adapter; and

sending means for sending the new data packet to the target destination.

27. The data processing system of claim 15, further comprising means for adding, during internal routing of the data packet to a given partition of the logical partitioned data processing system, the first flag and the second flag to a header of the data packet that is received.

28. The data processing system of claim 15, wherein the first flag and the second flag are located in a header in the data packet.

29. A computer program product recorded on a computer readable, recordable-type medium and operable for processing a data packet in an interpartition virtual network by a logical partitioned data processing system, wherein the logical partitioned data processing system comprises a plurality of logical partitions that are each operable for simultaneously running an operating system therein, wherein each one of the logical partitions comprises a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical partitioned data processing system to thereby form the interpartition virtual network, the computer program product comprising:

first instructions, responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, for identifying a state of a first flag and a state of a second flag in the data packet; and

second instructions for selectively verifying a checksum, by the first partition in the logical partitioned data processing system, for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet.

30. The computer program product of claim 29, wherein the first instructions comprise sub-instructions for (i) determining, using the first flag, whether or not there is a checksum value included in the data packet that is received and (ii) determining, using the second flag, whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received.

31. The computer program product of claim 30, wherein the second instructions includes: sub-instructions for verifying the checksum for the data packet only if the first flag and the second flag are unset.

32. The computer program product of claim 30, wherein the second instructions includes: sub-instructions for skipping verification of the checksum if the first flag is set.

33. The computer program product of claim 30, wherein the second instructions includes: sub-instructions for skipping verification of the checksum for the data packet if the first flag is unset and the second flag is set.

34. The computer program product of claim 30, further comprising instructions to conditionally unset the second flag if the packet was received through a first virtual network adapter associated with the first partition.

35. The computer program product of claim 30, further comprising instructions to conditionally set the first flag if the data packet, received from the second partition, originated from within the logical partitioned data processing system.

36. The computer program product of claim 30, further comprising instructions to conditionally unset the second flag if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked.

37. The computer program product of claim 36, further comprising instructions to conditionally unset the first flag and conditionally unset the second flag if the data packet was received by a physical adapter that (i) is associated with the second partition, and (ii) does not support checksum offload.

38. The computer program product of claim 36, further comprising instructions to conditionally unset the first flag and conditionally set the second flag if a physical network adapter, supporting a checksum offload, verified the checksum as being good.

EVIDENCE APPENDIX

This appeal brief presents no additional evidence.

RELATED PROCEEDINGS APPENDIX

This appeal has no related proceedings.